

优化生鲜商超收益—蔬菜商品的定价与补货决策分析

摘 要

蔬菜商品在生鲜商超中占据大部分销售额，但其受时间影响因素大，故其定价与补货决策具有重要意义。本文通过建立优化模型，从蔬菜品类及单品相互关系入手，逐渐与实际情况相融合，对蔬菜商品定价与补货决策进行深入分析。

针对问题一，一是分析蔬菜各品类及单品销售量的分布规律，二是分析蔬菜各品类及单品销售量的相互关系。本文通过**数据可视化**的方法对问题一中的分布规律进行分析，运用 **python** 绘制蔬菜各品类及单品的折线图。得知包括在 2020-2023 年，花叶类、辣椒类、食用菌这三类蔬菜销量是 6 个品类中较高的等分布规律。引入 **person 相关系数** 对指标之间的关系判断，得知包括食用菌、辣椒类、水生根茎类三者销量相互影响较大且呈正相关等相互关系，进而分析商超在营销策略方面的捆绑销售等特点。

针对问题二，将问题分为两个部分：一为各蔬菜品类销售总量与成本加成定价的关系，二是商超收益最大化下给出未来一周各蔬菜品类的补货总量与定价策略。在研究前者时，通过绘制各蔬菜品类销量与价格的**散点图**，观察发现品类销量与价格呈一定关系。在研究后者时，通过**线性回归模型**和**多目标规划**在符合成本加成定价的同时，求得各蔬菜品类的日补货总量与定价策略。

针对问题三，本文基于问题二**多目标规划模型**引入 **0-1 整数规划**思想，建立补货与定价决策的综合规划模型，其中 **0-1 整数规划模型**简单表示单品销售情况，由于该模型运算量较大，故利用**多目标粒子群优化算法**，确定单目标的最优解后找到全局最优解。在未来一周使商超收益最大化时，30 个单品蔬菜的日补货总量与定价策略。

针对问题四，为了商超更好地指定蔬菜商品的补货和定价决策，商超主要需要采集两大方面的相关数据，一方面是**顾客信息**，另一方面是**商品信息**。本文还建立了一个商超**库存与销售**的优化模型，帮助商超合理规划库存。计算出蔬菜各品类需要的订货量和最大缺货量，减少因库存挤压等原因导致商超的经济损失和食品浪费。

上述研究对生鲜商超收益有建设性影响。

关键词：数据可视化；线性回归；多目标规划模型；多目标粒子群优化算法

一、 问题重述

1.1 问题背景

蔬菜类商品的定价与补货决策对于生鲜零售业至关重要。因蔬菜保鲜期短、品相易受时间影响，需及时调整定价与补货以确保商品新鲜质量和减少浪费。

在生鲜商超中，故需每天进行补货和定价决策。商超采用"成本加成定价"方法，但需根据销售和需求情况制定补货计划。此外，不同品类和单品的销售量之间可能存在关联关系，而销售空间有限，需合理控制销售组合。

1.2 问题重述

因蔬菜保鲜期短、品相易受时间影响，需及时调整定价与补货以确保商品新鲜质量和减少浪费。因此，商超通常会根据各商品的历史销售和需求情况每天进行补货。题目共给出了 4 个附件数据，根据附件相关数据建立模型，解决以下四个问题：

针对问题一：分析蔬菜品类和单品的销售趋势，并探究它们之间的关联性，在考虑蔬菜类商品销售管理时，了解销售趋势以及不同商品之间的关联性具有重要学术价值。

针对问题二：基于历史销售数据和成本加成定价信息，制定 2023 年 7 月 1 日-7 日未来一周的品类补货计划和定价策略，以最大化商超收益。在蔬菜类商品的供应链管理中，制定科学合理的品类补货计划和成本加成定价策略对于商超的经济效益至关重要。

针对问题三：在可售品种数量控制在 27-33 个、可售品种范围控制在 6 月 24 日-30 日及每个单品最低陈列量为 2.5 千克的条件下，制定 7 月 1 日的单品补货计划和定价策略，以最大化提高商超经济效益。

针对问题四：商超需要采集相关数据，以便提高蔬菜商品销售优化决策的准确性和可靠性。

二、 问题分析

2.1 问题一的分析

问题一中，蔬菜类商品不同品类或不同单品之间可能存在一定的关联关系，分析蔬菜各品类及单品销售量的分布规律及相互关系。

对于分布规律：蔬菜各品类及单品销售量的分布规律，首先在分析本问题时通过对附件 2 中的销售流水明细数据进行预处理，以便后续的分析 and 建模。对销售流水明细数据进行合并，得到各蔬菜品类和单品的销售总量，利用数据预处理得到的数据集，绘制不同品类或不同单品之间的散点图，分析各蔬菜品类与单品销售量从月维度上研究具体变化规律。

对于相关关系：对于蔬菜各品类及单品销售量的相互关系，是评价模型中的属于关联分析。首先，利用 K-S 检验帕丁数据服从正态分布，计算出 6 个蔬菜品类的商品信息 Person 相关系数并绘制蔬菜各品类相关系数热力图进行分析。

2.2 问题二的分析

问题二中，需对蔬菜各品类进行的销售总量与成本加成定价的分析，得出2023年7月1日-7日未来一周蔬菜总量的补货总量和定价决策。

根据上述拆分成两部分问题，本题需先对蔬菜各品类销量 B 、销售单价 a 和批发单价 b 建立线性回归模型进行预测，并在通过附件中的数据得出各蔬菜品类的损耗率 t 的条件下，建立多目标规划模型得出未来一周的供应量 A 即蔬菜各品类补货总量与加成率 w 。

通过数据预处理得到的数据集运用 python 以茄类为例绘制出销量与销售单价即成本加成定价的散点图，得销量与销售单价呈一定关系。接着，以日补货总量为决策变量、商超收益为目标函数，构建优化模型，对日补货总量及定价策略进行求解。

2.3 问题三的分析

问题三中，针对于2023年7月1日的单品补货量和定价决策，与问题二具有相似的建模过程和求解。

由于在商超中，蔬菜类商品的销售及存放空间有限，可售单品数规定在 $[27,33]$ 区间上，且各单品订购量至少满足2.5kg。因针对于7月1日，可售单品考虑范围在6月24日-30日之间。

基于问题二中多元线性回归模型的基础上，综合0-1整数规划，增加决策变量，其值为1时表示某个蔬菜单品进行出售，其值为0时则表示没有进行出售。通过上述改进后的模型，使用多目标粒子群优化算法进行求解，得出7月1日的单品补货量和定价决策。

2.4 问题四的分析

问题四中，为了更好地制定蔬菜商品的补货和定价决策，商超还需要采集顾客信息和商品信息这两大相关数据以更好地规划补货计划和定价策略。首先画出了6个蔬菜品类的商品信息相关系数热力图，先分析各蔬菜品类之间的关系。本文分别为问题提供了具体的顾客信息和商品信息的相关数据，接着对于商品信息建立了一个简单的商场库存与销售的优化模型，帮助商超合理规划库存。

三、 模型假设

1. 销售数据准确、完整，没有重复或错误的记录。
2. 历史销售数据可以用来预测未来销售趋势。
3. 销售数据相对稳定，不受短期内的突发事件（如自然灾害）的显著影响。
4. 将提供的销售数据视为准确且完整，不包含任何误差或遗漏。
5. 季节性因素可能会影响蔬菜销售，如供应量、天气条件或节假日等
6. 市场上的竞争对手行为相对稳定，价格策略和促销活动不会剧烈波动。

四、 符号说明

符号	说明	单位
S	销售总量	元
M	成本	元
w	加成率	%
t	损耗率	%
A	供应量	Kg
B	销量	Kg
a	销售单价	元/Kg
b	批发单价	元/Kg
W	利润	元

五、 数据预处理

5.1 异常值处理

由问题一题意得, 本题对蔬菜各品类及单品销量的分布规律和相互关系进行分析, 为统计方便忽略销售类型为退货的样本数据。

先对于给出的数据进行缺失值查找, 通过 python 的 `numpy.isnan()` 函数寻找空缺值, 得出附件数据不存在缺失值, 后进行异常值判定及处理工作。

再进行异常值查询, 通过 Excel 表格筛选功能, 观察给出的数据, 我们发现给出的数据中存在逻辑问题异常值, 如下所示:

编码为 102900011034354, 该商品为鲜粽叶(袋)(1), 在 2022-06-09 销售数量为 160 千克, 与实际情况有很大出入, 因此可以认定为异常数据。详细数值如下所示:

表 1 人为判定异常值

销售日期	扫码销售时间	单品编码	销量(千克)	销售单价(元/千克)
2022-06-09	09:31:57.045	102900011034354	160.000	5.90

3 σ 原则判定异常值, 当给出的数据集服从正态分布时, 有 99.7% 的数据落在均值的三倍标准差范围内, 如图 1 所示。对于一个数据点落在三倍标准差之外, 概率仅为 0.3%, 为小概率事件, 故将该数据认定为异常值。异常值 3 σ 判定公式如下所示:

$$P(|x - \mu| > 3\sigma) \leq 0.003 \quad (1)$$

其中, x 是收集的数据集中的一个数据点, μ 是 x 对应这一指标下所有数据的均值, σ 是 x 对应这一指标下所有数据的标准。

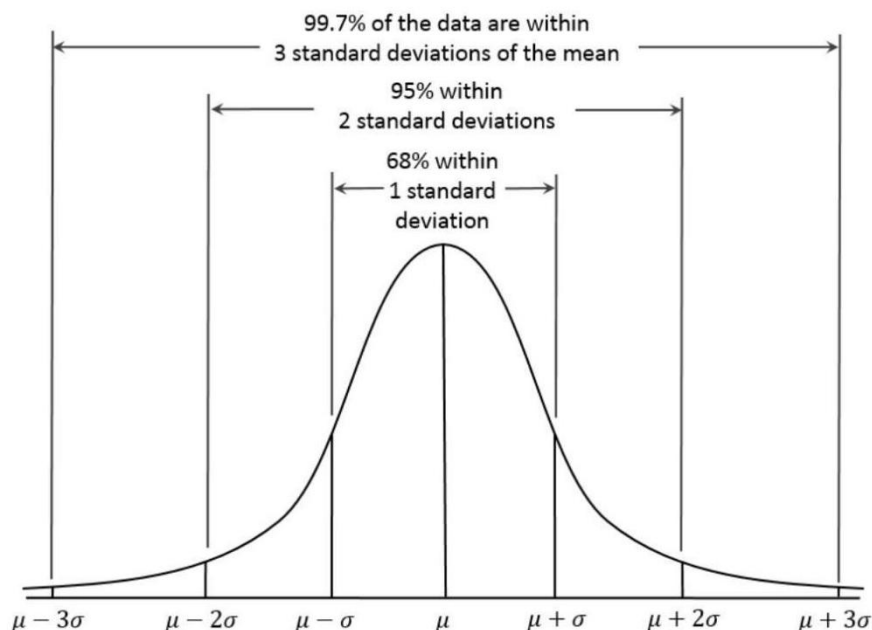


图 1 3 σ 异常值规则示意图

首先，对相关指标进行正态性检验。如果数据服从正态分布，将应用 3 σ 原则来识别异常值，随后进行人工审核以降低假阳性的风险。对于不符合正态分布的数据集，直接采用手工判断流程。在该过程中，数据将按升序排列，同时结合领域知识和商业背景进行深入分析，以识别和处理可能的分类错误。进一步审查和处理误分类数据，以进行全面分析和解决，

在对收集的数据集进行异常值检测和处理后，考虑到样本之间的独立性，确定所有样本都被分类为异常值，并随后被移除。最终，异常值的识别基于缺失值、逻辑一致性评估和应用 3 σ 原则。使用 Excel，所有已识别的异常值都被标记并标记为数字 '1'。利用 Excel 的筛选功能，分离了标有 '1' 的数据点，结果如图 5 所示。共累计识别了 1,767 个异常值。对于这 1,767 个异常值所在的各个样本，决定将它们排除在进一步的分析和处理之外。

利用 Python 通过 SciPy 库中的 kstest 函数计算的进行 KS 检验，输出结果包括 KS 检验统计量 D 值和 p 值。其中，D 值越小则说明数据越接近正态分布，越大则说明数据偏离正态分布；p 值越小则越有可能拒绝原假设，当 p 值>0.05 时说明待检验数据符合正态分布。

表 2 蔬菜各品类 D 值、p 值

蔬菜品类	水生根茎类	花叶类	花菜类	食用菌	辣椒类	茄类
D 值	0.135	0.189	0.165	0.190	0.204	0.190
p 值	0.960	0.716	0.849	0.661	0.630	0.778

蔬菜各品类 D 值、p 值如表 2，故 6 个蔬菜品类整体上符合正态分布。

5.2 数据合并

先把附件 1 和附件 4 用 excel 进行合并进行合并，再以单品名称为标识符与附件 2 合并，最后以销售日期和单品编码为标识符与附件 3 进行合并，另外将销

量（kg）和销售单价（kg/元）相乘得到销售量这一新列，完成数据合并。以便后续进行问题建模和分析，完整数据文件因大小问题，故转化为 txt 文件并压缩见支撑材料。

六、 问题一

6.1 针对问题一中分布规律

为了更加直观的观察蔬菜各品类之间的销售量的变化，绘制如图 1，其中横坐标从 2020 年 7 月开始：

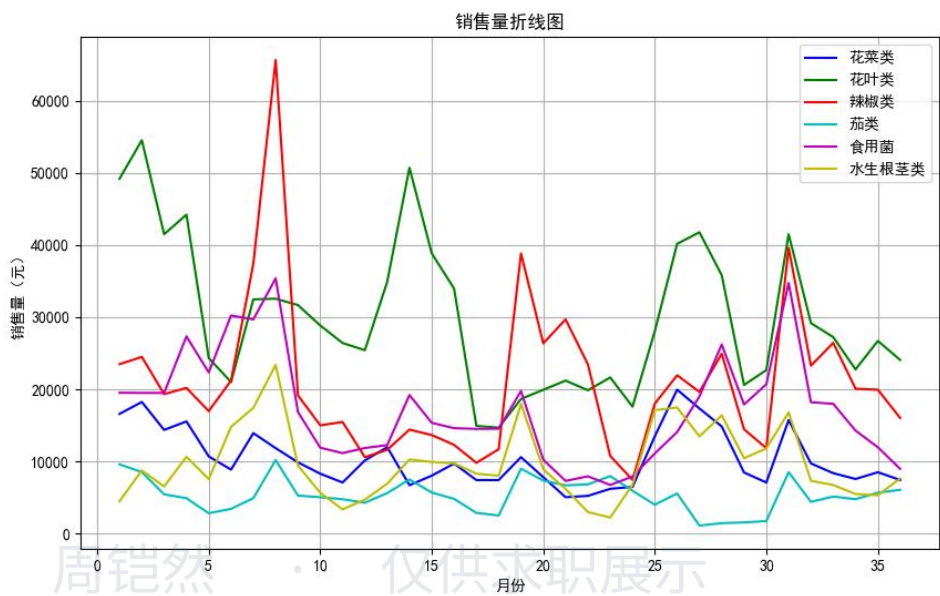


图 2 蔬菜各品类 2020 年 7 月-2023 年 6 月每月销售总量示意图

由图明显看出 2020-2023 中花叶类、辣椒类、食用菌蔬菜销量较高，且该三类有明显峰值。花菜类、茄类、水生根茎类销售总量换较为平稳且销量较低。

在 2021 年 2 月（即图中第 8 月）、2022 年 2 月时且除花叶类以外 5 类蔬菜均出现峰值，而花叶类峰值出现在 2021 年 2 月、2022 年 2 月前 4-5 个月左右，2023 年 2 月 6 类蔬菜均出现峰值。2022 年 8 月-2023 年 6 月，6 类蔬菜间的销售总量差异明显小于 2020 年 6 月-2022 年 7 月之间。

蔬菜各品类呈现阶段性变动规律、周期性波动特征、季节性波动规律、非对称性特征^[1]：

阶段性变动规律即某一段时期与其前期和后期相比变动趋势相反；周期性波动特征即将每年 6 月到次年 6 月视为一个周期，各周期内月维度销售总量跌时间多于上涨时间；季节性波动规律即月维度销售总量在第一季度时从峰值逐渐下降，第二季度时趋于平缓，第三季度开始逐渐上升，第四季度进入快速上升期；非对称性特征即不同年内各月份间分布规律不尽相同。

经多年的建设发展，蔬菜流通形成“南菜北运”“北菜南销”全国性流通大格局，建立了以农批市场为节点现代流通体系，故近年来蔬菜始终处于供大于求状态^[2]，

根据经济学上经典的供需曲线可知，供大于求时市场销售总量呈下降趋势，符合2020年7月-2023年6月6类蔬菜多数各品类销售额总体呈下降趋势。

6.2 针对问题一中相互关系

在分析蔬菜各品类的相互关系时，我们引入 person 相关系数，进行指标之间关系的判断。

样本协方差、标准差和皮尔逊相关系数公式如下，其中 n 为样本数据点的数量， X_i ， Y_i 为两个变量的第 i 个观察值：

Sample Covariance (S) = $\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$ (1)

Sample Standard Deviation (S) = $\sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}$ (2)

Pearson Correlation Coefficient (r) = $\frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$ (3)

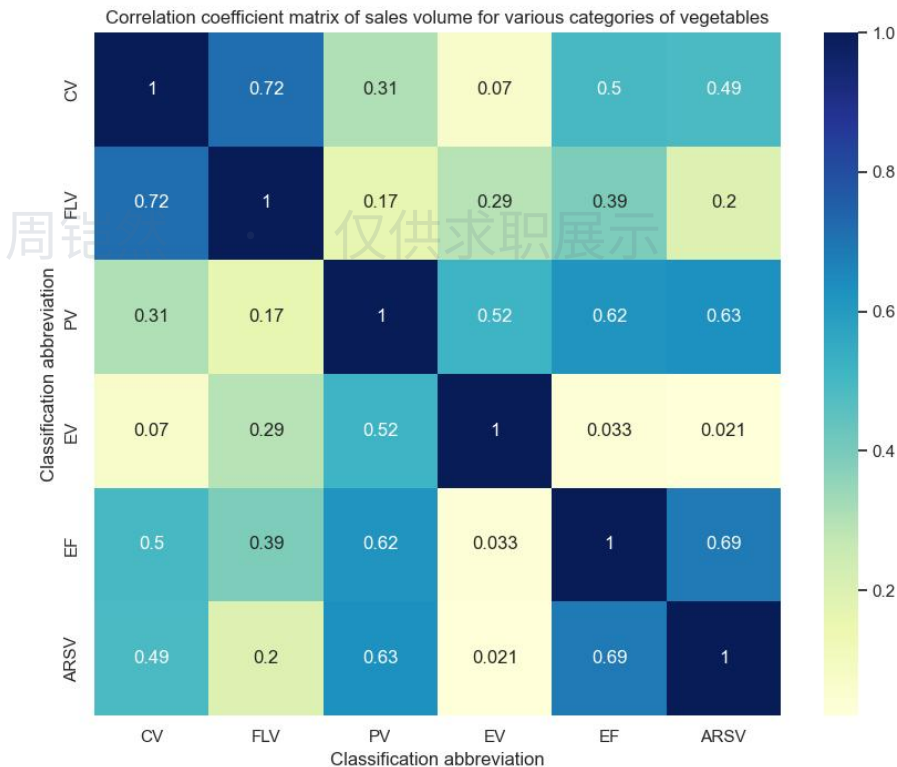


图3 蔬菜各品类相关系数热力图

故得蔬菜6品类的相关系数热力图如上，食用菌、辣椒类、水生根茎类三者销售量相互影响较大且呈正相关，花菜类和花叶类、茄类和辣椒类同样相互影响较大且呈正相关。

食用菌、辣椒类和水生根茎类三者在销售量上存在显著的正相关性。季节性

需求方面，观察到它们可能在不同季节呈现相似的销售趋势。例如，在寒冷的冬季，人们更倾向于购买食用菌和辣椒类蔬菜，而在潮湿的季节，水生根茎类蔬菜可能更受欢迎。配伍性方面，这三者在中国菜肴中常常一同使用，可能导致三者之间的销售量相互影响。

花菜类和花叶类在季节性需求方面，都属于叶菜类别，故可能具有相似的季节性需求；茄类和辣椒类尤其在季节性需求方面，其所适应的气候条件相似，都具有耐热的特性，更适合在夏季销售。

在营销策略方面，蔬菜市场常有捆绑销售的特点，故造成上述各品类间的相关性。

七、 问题二

7.1 各蔬菜品类的销售总量与成本加成定价的关系

为了方便分析各蔬菜品类的销售总量与成本加成定价的关系，本文以茄类的销售与价格之间的关系为例进行具体分析，其他类别的蔬菜的相关信息图见支撑材料。

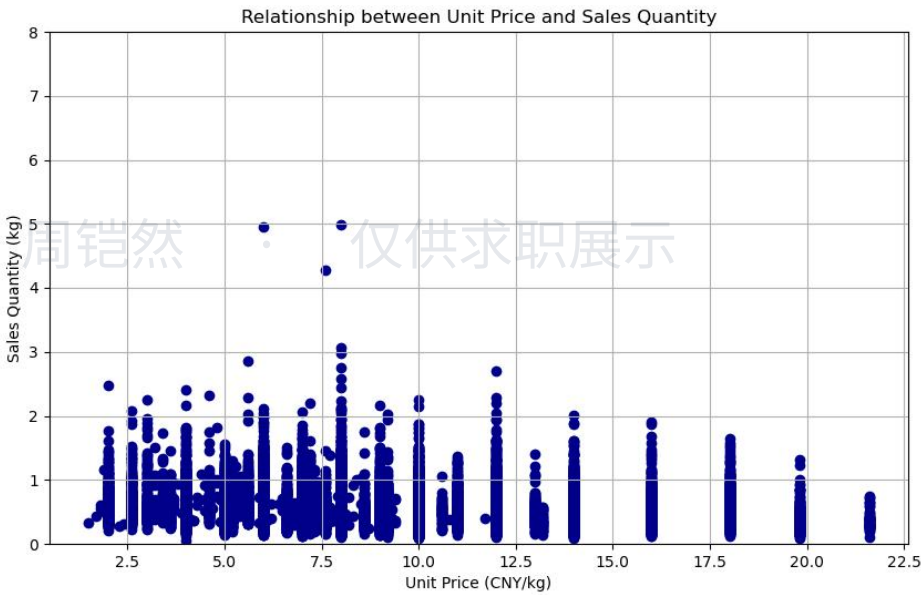


图 4 茄类销量与销售单价关系图

由上图可以看出茄类销售单价为范围在 2.5 元/kg 到 10 元/kg 之间保持平稳的关系，销售数据在短期内是相对稳定的。这一信息对于商超来说，可以根据这一信息制定更合理的定价策略。当茄子的单价设置在一个较为合理的价格范围内时，销量保持稳定。

对商超而言，可以在保持销售量的同时获得合理的利润；在茄类销售单价大于 10 元/kg 后，销量随着销售单价的增加而逐渐减小，后续两者呈现负相关的关系。高价格会使顾客对购买茄类感到犹豫，导致销量下降。顾客可能会寻找替代品或者减少购买量，从而影响商超的销售业绩。

7.2 未来一周的日补货总量和定价策略

线性回归分析是数据分析中最基础也是最重要的分析工具，绝大多数的数据分析问题，都可以使用回归的思想来解决。回归分析的任务是研究自变量和因变量的相关关系，尝试去解释的形成机制，进而达到通历史数据预测因变量的目的。

$$y_i = \beta_1 + \beta_0 x_i$$
$$\xi \sim N(0, \sigma^2)$$
(1)

式中， ξ 服从正态分布， β_1 为自变量系数， β_0 为常数项系数。 y_i 表示花菜第 i 天的定价， x_i 表示第 i 天某类蔬菜的总销售量。

由线性回归过渡到多元线性回归模型，多元线性回归指的是一个样本有多个特征的线性回归问题。对于一个有 n 个特征的样本 i 而言，它的回归结果如下方程：

$$\hat{y} = w_0 + w_1 x_{i1} + w_2 x_{i2} + \dots + w_n x_{in}$$
(2)

在上方这个表达式中，被统称为模型的参数，其中被称为截距(intercept)， $w_0 \sim w_n$ 被称为回归系数(regression coefficient)，有时也用表示。其中是目标变量， $x_{i1} \sim x_{in}$ 是样本 i 上的不同特征。

如果考虑有 m 个样本，是包含了 m 个全部样本的回归结果列向量，我们可以使用矩阵来表示这个方程，其中可以被看做是一个结构为(1,n)的列矩阵，是一个结构为(m,n)的特征矩阵，则有：

$$\begin{bmatrix} \hat{y}_0 \\ \hat{y}_1 \\ \hat{y}_2 \\ \dots \\ \hat{y}_m \end{bmatrix} = \begin{bmatrix} ccccc1 & 1 & 1 & \dots & 1 \\ x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & x_{m3} & \dots & x_{mn} \end{bmatrix} * \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \dots \\ w_n \end{bmatrix}$$
(3)
$$\hat{y} = Xw$$

线性回归的任务，就是构造一个预测函数来映射输入的特征矩阵 X 和标签值 y 的线性关系，而构造预测模型的核心就是找出模型的参数向量 w 。在多元线性回归中，我们将损失函数定义为：

$$\sum_{i=1}^m (y_i - \hat{y}_i)^2$$
$$\sum_{i=1}^m (y_i - X_i w)^2$$
(4)

损失函数衡量了我们构造的模型的预测结果和真实标签的差异，我们也固然希望预测结果和真实值差异越小越好，所以我们的求解目标就可以转化成：

$$\min |y - Xw|_2^2$$
(5)

而这未来一周的供应量可由上述预测的销量所得如下式， A 为供应量， B 为销量， t 为某品类的损耗率：

$$A = \frac{B}{1-t}$$
(6)

$$\frac{\sum_{i=1}^n \text{单品供应量} \times t_i}{\sum_{i=1}^n \text{单品供应量}} \quad (7)$$

最小二乘法是为了求解让残差平方和最小化的参数向量 w ，通过最小化真实值和预测值之间的残差平方和来求解参数。最终得到参数 w 最优解为下式：

$$w = (X^T X)^{-1} X^T y \quad (8)$$

我们通过 python 中的 numpy 函数得预测出 2023 年 7 月 1 日-7 日的 6 类蔬菜销量如下表：

表 3 2023 年 7 月 1 日-7 日各类蔬菜销量预测值

蔬菜品类 销量	水生根茎类	花叶类	花菜类	食用菌	辣椒类	茄类
7 月 1 日	45.95	164.72	27.391	50.73	99.211	8.791
7 月 2 日	46.89	162.58	26.43	49.90	100.85	7.73
7 月 3 日	46.32	159.97	23.39	46.53	101.62	4.63
7 月 4 日	54.53	164.74	28.37	51.94	108.21	9.37
7 月 5 日	57.95	155.26	20.14	42.64	102.21	2.35
7 月 6 日	55.64	157.50	22.45	46.02	107.73	5.62
7 月 7 日	59.62	158.31	23.84	47.35	111.13	5.85

7.2.1 销售量数据趋势分析

首先绘制出各蔬菜品类的销售量在不同时间点（月份）的趋势图。分析销售量趋势，可以为商超制定未来一周的蔬菜补货计划和定价策略提供以下三点有用的信息，帮助商超理解不同蔬菜品类的销售情况。

1.识别销售的最高值和最低值，从而更好地规划未来一周的补货计划和定价策略。商超可以更准确地预测哪些蔬菜品类可能在未来一周销售增加或减少，以便做出最优决策，以最大程度地提高收益。

2.了解销售趋势可以帮助商超决定每个蔬菜品类的最佳库存水平。如果某个品类的销售在未来一周预计会增加，商超可以增加库存以满足需求。

3.销售趋势图提供了关于蔬菜销售的历史数据，商超可以根据这些数据来预测未来一周不同蔬菜品类的销售量。这有助于商超合理规划补货计划。

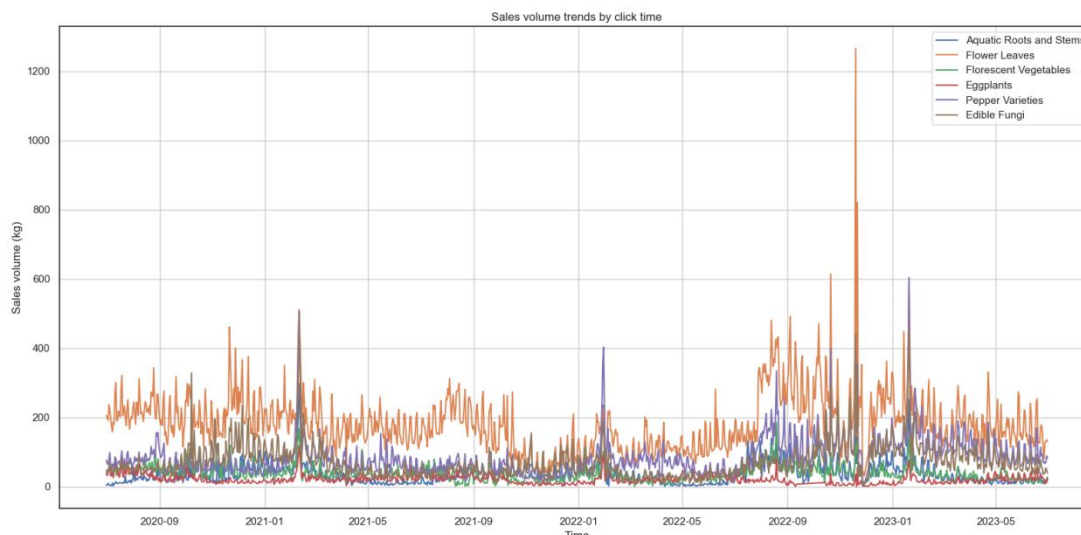


图 5 蔬菜各品类时间与销量趋势图

7.2.2 针对问题二规划模型的建立与求解

对于各蔬菜品类未来一周的日补货总量和定价策略建立多目标规划模型求解，全部蔬菜品类最终将展示计算结果于附录。

令决策变量为 x_{ij} , 其为第 i 天 j 类蔬菜品类销售总量，由于定价采用成本加成定价，成本加成定价是一种定价策略，以确定产品或服务的销售单价，该定价方法的核心思想是将成本与所需的利润相结合，来确认最终销售单价。

成本加成定价法有三大优点: (1) 定价简单，不需要历史边际成本和边际收入信息，便于操作，生产商和顾客均易理解，有利于产销各方的合作。(2) 价格稳定。通常市场中，升高价格会招致顾客的抵制，降低价格则会带来竞争企业的恶性价格竞争，而成本加成定价法有利于价格稳定。(3) 便于价格领导。成本加成定价法向竞争者传达的信息是成本增加需要提高价格，有利于合作定价。这对生产企业是极其有利的^[4]。

故设每日加成率为 w_{ij} , 最终定价 y_{ij} 可表示为:

$$y_{ij} = (1 + r_{ij})c_{ij} \quad (1)$$

其中， c_{ij} 花菜表示 i 日某类蔬菜的进价成本。因此，对于该类第 i 日的利润可表示为:

$$W_{ij} = r_{ij} * x_{ij} \quad (2)$$

因此，2023 年 7 月 1 日-7 日内商超的总盈利可表示为:

$$W_{\text{总}} = \sum_{i=1}^7 (w_{i\text{菜}} + w_{i\text{叶}} + w_{i\text{辣}} + w_{i\text{菌}} + w_{i\text{茄}} + w_{i\text{水}}) \quad (3)$$

对于约束条件，主要约束销售量与定价的关系:

$$\begin{aligned} y_{i\text{菜}} &= a_{10} + a_{11}x_{i\text{菜}} \\ y_{i\text{叶}} &= a_{20} + a_{21}x_{i\text{叶}} \\ y_{i\text{辣}} &= a_{30} + a_{31}x_{i\text{辣}} \\ y_{i\text{茄}} &= a_{40} + a_{41}x_{i\text{茄}} \end{aligned} \quad (4)$$

$$y_{i \text{ 菌}} = a_{50} + a_{51}x_{i \text{ 菌}}$$

$$y_{i \text{ 水}} = a_{60} + a_{61}x_{i \text{ 水}}$$

最终得茄类结果如下表：

日期	日补货总量 (kg)	定价策略 w (%)
2023.7.1	9.791	96.00
2023.7.2	8.732	41.80
2023.7.3	5.622	64.84
2023.7.4	10.835	105.51
2023.7.5	3.295	95.34
2023.7.6	6.624	109.11
2023.7.7	6.153	84.10%

7.3 问题结论

通过上述分析，价格和销量之间关系密切，可拟合为一元线性方程，具有较强的线性负相关关系。且当价格增高时，销量减少；当价格低时，销量增加。这一规律与目前农产品普遍存在增产不增收现象表现一致^[3]。

运用蔬菜各品类多元线性回归模型，能够预测与价格具有强相关性的销量数据，这对市场价格调控、辅助菜商菜农经营等起到较大积极作用。

八、 问题三

8.1 基于问题二模型的建立与求解

问题三由题意与问题二背景下，可运用 0-1 整数规划来解决。0-1 整数规划是一种优化问题，其中每个决策变量的取值限定为 0 或 1。在你的情况下，每个蔬菜单品的出售与否表示为一个 0-1 整数变量，其中 0 表示不出售，1 表示出售。

决策变量：对于 251 种蔬菜单品，引入 251 个 0-1 整数变 q_i ($i=1,2,3,\dots,251$) 其中 q_i 等于 1 表示该类蔬菜单品出售，等于 0 表示无出售记录。

因此，可以用 q_i 开关变量求和来表示单品总数控制约束。即：

$$27 \leq \sum_{i=1}^{251} q_i \leq 33 \quad (1)$$

对于各单品订购量满足最小陈列量约束，即可以表示为每种单品总量 e_i ， $i=(1,2,3,\dots,251)$ 需要大于 2.5 千克，可表示为：

$$e_i \geq 2.5, \quad i=1,2,3,\dots,251 \quad (2)$$

关于利润的计算，沿用问题二中对成本加成定价的理解，故最终总盈利表示为：

$$W_{\text{总}} = \sum_{i=1}^{251} w_i x_i \quad (3)$$

故得问题三建立的优化模型如下所示：

$$\max W_{\text{总}} = \sum_{i=1}^{251} q_i w_i x_i \quad \left\{ \begin{array}{l} y_i = (1 + w_i) c_i \\ y_i = k_i x_i + b_i \\ 27 \leq \sum_{i=1}^{251} q_i \leq 33 \\ e_i \geq 2.5 \\ q_i = 0 \text{ or } 1 \\ c_i, x_i \geq 0 \\ w_i, y_i \in R \\ i = 1, 2, 3, \dots, 251 \end{array} \right. \quad (4)$$

由于问题三建立的规划模型直接进行求解时，面对巨大的运算量，故对于我们建立的多元规划模型，利用多目标粒子群优化算法进行求解。

粒子群算法最早在 1995 由 Eberhart 和 Kennedy 提出，其特点时简单易行，收敛速度快，设置参数少。这些特点正好巧妙地弥补了我们建立模型的缺陷。我们基于传统的粒子群算法，结合我们的这个题目进行改进。通过我们的规划模型，每个粒子确定每个粒子个体的最优解并从这些个体最优解找到一个全局最优值，这里涉及到了我们粒子的位置和距离公式^[5]。

$$z_{ij} = z_{ij} + d_{1ij} * \text{rand}() * (pbes_{ij} - x_{ij}) + d_{2ij} * \text{rand}() * (pbes_{ij} - x_{ij}) \quad (5)$$

$$x_{ij} = x_{ij} + z_{ij}$$

其中，rand() 为 0 到 1 之间随机数。

$$z_{ij} = O * z_{ij} + d_{1ij} * \text{rand}() * (pbes_{ij} - x_{ij}) + d_{2ij} * \text{rand}() * (pbes_{ij} - x_{ij}) \quad (6)$$

O 为惯性因子，其值非负。数值越大，则寻找最优解的能力越强，局部寻找最优解的能力越弱，反之结论正好相反。我们这里采用线性递减权值策略。

$$O^{(t)} = (O_{\text{ini}} - O_{\text{end}})(G_k - g) / G_k + O_{\text{end}} \quad (7)$$

G_k 为最大迭代次数，O_{ini} 为初始惯性权值，O_{end} 迭代到最大进化数时的惯性权值。

得到问题三结果部分如下所示，完整结果见附录：

单品编码	日补货总量 (KG)	定价策略 (%)	单品编码	日补货总量 (KG)	定价策略 (%)
102900005115625	3.172	111.79%	102900011035078	2.762	102.07%
102900005115779	10.443	134.23%	102900011035740	1	85.37%
102900005115878	9.014	142.23%	102900011021842	7	71.14%

九、 问题四

9.1 相关性热力图分析

根据相关性热力图中不同方块颜色对应的相关系数的大小,可以判断出变量之间相关性的。两个变量之间相关系数的计算公式为:



图 6 六大类蔬菜相关系数矩阵图

本题将蔬菜划分成了六类进行分类讨论。为了更好地指定蔬菜商品的补货和定价策略,商超还要从以下两大方面采集信息:

(一) 顾客信息

1. 人口统计信息: 在市场细分模型中, 使用聚类算法来将顾客分组, 以便于制定不同定价策略。可使用线性回归表示不同人口统计信息对价格敏感度的关系。以便制定适合不同消费者群体的定价策略。例如, 可以为高收入群体提供高端蔬菜品种, 并在定价上提供不同的选择。。

2. 购买偏好和消费行为: 概率分布来描述不同蔬菜品种的需求分布。例如, 可以使用概率密度函数来建模不同蔬菜的购买频率和购买数量。了解顾客的购买偏好, 包括他们对不同蔬菜品种的需求、购买频率、购买数量等信息, 这是可以帮助商超根据不同消费群体的需求指代更具吸引力的销售组合。购买历史和忠诚度: 了解顾客的购买历史和忠诚度可以帮助商超制定奖励计划或促销活动, 以提高顾客忠诚度并增加销售。

3. 购买历史和忠诚度: 了解顾客的购买历史和忠诚度可以帮助商超制定奖励计划或促销活动, 以提高顾客忠诚度并增加销售。

(二) 商品信息

1. 库容数据: 包括不同商品的储存需求, 即蔬菜的保鲜条件和库容限制。这有助于优化库存管理和空间利用。采购时间和批次信息: 了解商品的采购时间和批次可以帮助商超了解供应链的稳定性和季节性波动。这可以用于优化库存管理和补货计划。

2. 商品成本数据：包括蔬菜的采购成本、运输成本、储存成本等。这些数据对定价决策非常重要，商超需要了解成本以确保定价策略盈利。

3. 采购时间和批次信息：了解商品的采购时间和批次可以帮助商超了解供应链的稳定性和季节性波动。这可以用于优化库存管理和补货计划。

4. 商品质量和品相数据：维护商品质量和品相信息的数据库可以帮助商超监控商品的新鲜度和外观质量。这有助于定价策略和管理库存。

了解上述数据可以让商超更好地预测需求、计划补货、调整价格，并提高竞争力。通过数据分析，商超能够做出更明智的决策，满足客户需求，提高利润和效率。

9.2 商场库存与优化的模型

1. 决策变量：

$$\begin{aligned} Q_i &: \text{第} i \text{种商品的订货量} \\ C_i &: \text{第} i \text{种商品的采购成本} \\ S_i &: \text{第} i \text{种商品的库容需求} \end{aligned} \quad (1)$$

2. 目标函数：最小化总成本，包括订货成本和库容成本

$$\text{Minimize } \sum_{i=1}^n (C_i \cdot Q_i + S_i \cdot H \cdot Q_i) \quad (2)$$

其中，H 是单位库容的成本。

3. 约束条件：订货量不可为负数；总订货成本不能超过可用资金(假设总可用资金为 B)；总库容需求不能超过库容限制(假设总库容限制为 S_{total}):

$$Q_i \geq 0, \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^n (C_i \cdot Q_i) \leq B \quad (3)$$

$$\sum_{i=1}^n (S_i \cdot Q_i) \leq S_{\text{total}}$$

本文所建立的商超库存与销售的优化模型，帮助商超合理规划库存，确保有足够的商品供应以满足客户需求，同时避免过多的库存积压，减少损失和浪费。通过合理的供应链管理和定价策略，商超可以在竞争激烈的市场中取得更大的成功。

十、模型的评价及推广

10.1 模型的优点

1) 在数据预处理过程中，进行了缺失值的检查并排除了存在逻辑问题的样本数据，以确保模型输入数据的质量，从而提高了模型可信度。

2) 在统计分析和关联分析阶段，我们运用散点图、D 值、p 值、以及 Pearson

相关系数等多种统计方法，深入研究销售数据的分布规律和相互关系。这一系列方法有助于理解数据的特性和相关性。

3) 问题二和问题三的优化模型，将商超的总收益作为目标函数，将每日补货总量和定价策略作为决策变量，并考虑所有相关约束条件，上述以寻找最优的补货计划和定价策略为目标，以最大程度地提高商超的盈利能力。

4) 模型简单易懂，方法灵活，具有较强的推广性。

10.2 模型的不足

1) 使用 D 值和 p 值检验数据是否符合正态分布存在一些限制。这种方法对样本大小敏感，对于多峰分布的数据样本，无法提供具体分布参数，容易受离群值干扰。

2) 在异常值处理过程中，人工判断可能存在主观因素的干扰，不同的个体可能会得出不同的结论。为了减少主观性的影响，可考虑制定清晰的异常值处理规则或采用自动化的异常值检测方法，以提高处理的客观性和一致性。这有助于确保对数据异常的识别和处理具有更高的可信度和准确性。

3) 建立的模型应用于商超实际运营时，会遇到一些挑战。这包括数据集成、自动化更新和决策支持等方面的问题。需仔细考虑如何有效地将模型整合到商超的运营流程中，以确保其实际执行和有效性。还涉及到数据的顺利流通、模型的定期更新以适应变化的环境，以及如何将模型的结果用于支持决策制定和实施。

10.3 模型的推广

综合数据分析与优化模型是一个用于解决商超蔬菜商品补货和定价决策的强大工具。虽然模型包括多个复杂步骤，如数据预处理、统计分析、关联分析和优化建模，但其优点在于综合性、灵活性和支持决策的能力。该模型在提供深刻市场洞察的同时，还有助于商超更智慧地制定补货计划和定价策略，提高竞争力，最大化利润率。

该综合模型可适用于类似研究，并根据具体问题和对象进行定制和改进。模型中使用的方法和技术也具备推广到其他领域的潜力，但在推广时需要仔细考虑模型的适应性和适用范围，并进行验证和论证。

十一、 参考文献

- [1] 赵晓飞.蔬菜价格波动的规律、影响因素与调控对策研究[J].当代经济管理,2015,37(02):37-42.DOI:10.13253/j.cnki.ddjjgl.2015.02.008.
- [2] 刘金艳.大型突发事件背景下蔬菜市场实现供需紧平衡的分析及对策建议[J].商业经济,2023(08):101-103+196.DOI:10.19905/j.cnki.syjj1982.2023.08.017.
- [3] 杨皓旭.蔬菜价格与销量的相关性分析研究——以油麦菜为例[J].食品安全导刊,2018(21):179-181.DOI:10.16043/j.cnki.cfs.2018.21.146.
- [4] 韩俊华,干胜道.成本加成定价法评介[J].财会月刊,2012(22):74-75.DOI:10.19641/j.cnki.42-1290/f.2012.22.034.
- [5] 贺宇,赵建航,孔朝阳. 中国产品在东盟市场的印象研究——基于印尼、泰国、老挝三国调查问卷的分析[J].科学决策,2022(10):125- 137.

周铠然 · 仅供求职展示

附录

附录 1

介绍：支撑材料中

文件：四个附件合并后数据集

数据预处理：各个单品数据

数据预处理：各品类按月份分的销售额

等多个文件

代码：各个品类的 python 代码

合并数据 python 代码

六大类蔬菜相关系数

等多个代码

附录 2

介绍：问题一代码

```
1. import pandas as pd
2. import matplotlib.pyplot as plt
3. import numpy as np
4.
5. # 文件路径列表
6. file_paths = [
7.     "C:/Users/Desktop/分析结果/花菜类.xlsx",
8.     "C:/Users/Desktop/分析结果/花叶类.xlsx",
9.     "C:/Users/Desktop/分析结果/辣椒类.xlsx",
10.    "C:/Users/Desktop/分析结果/茄类.xlsx",
11.    "C:/Users/Desktop/分析结果/食用菌.xlsx",
12.    "C:/Users/Desktop/分析结果/水生根茎类.xlsx"
13. ]
14.
15. # 创建一个空的 DataFrame 来存储数据
16. data = pd.DataFrame()
17.
18. # 从每个文件中读取数据并添加到 DataFrame
19. for file_path in file_paths:
20.     class_name = file_path.split('/')[-1].split('.')[0] # 获取类别名称
21.     df = pd.read_excel(file_path)
22.     data[class_name] = df['销售量'] # 假设销售量的列名为"销售量"
23.
24. # 设置字体为黑体
```

```

25. plt.rcParams['font.family'] = 'SimHei'
26.
27. # 创建折线图
28. plt.figure(figsize=(10, 6))
29.
30. # 循环绘制每条折线，使用不同的颜色
31. colors = ['b', 'g', 'r', 'c', 'm', 'y']
32. x_values = np.arange(1, 37) # 横坐标从 1 到 36
33. for i, class_name in enumerate(data.columns):
34.     plt.plot(x_values, data[class_name], label=class_name, color=colors[i])
35.
36. # 设置图表标题和标签
37. plt.title('销售量折线图')
38. plt.xlabel('月份')
39. plt.ylabel('销售量 (元)')
40.
41. # 添加图例
42. plt.legend()
43.
44. # 显示图表
45. plt.grid(True)
46.
47. # 保存图表到指定路径
48. save_path = "C:/Users/Desktop/销售额折线图.png"
49. plt.savefig(save_path)
50.
51. # 显示图表
52. plt.tight_layout()
53. plt.show()
54.
55. print(f"图表已保存至 {save_path}")

```

附录 3

介绍：问题二代码

```

1. from sklearn.linear_model import LinearRegression
2. from sklearn.model_selection import train_test_split
3. from sklearn.metrics import mean_squared_error
4. import numpy as np
5.
6. # 根据日期和类别对销售量进行分组求和
7. daily_sales = merged_data.groupby(['销售日期', '分类名称'])['销量 (千
克)'].sum().reset_index()

```

```

8.
9. # 创建一个函数，使用简单线性回归模型预测给定类别的销售额
10. def forecast_sales(category):
11.     category_data = daily_sales[daily_sales['分类名称'] == category].reset_index(drop=True)
12.     category_data['day_num'] = np.arange(len(category_data))
13.
14.     x = category_data['day_num']
15.     y = category_data['销量(千克)']
16.
17.     # 将数据拆分为训练集和测试集（80%训练，20%测试）
18.     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, shuffle=False)
19.
20.     # 训练线性回归模型
21.     model = LinearRegression()
22.     model.fit(X_train, y_train)
23.
24.     # 预测测试集的销售额
25.     y_pred = model.predict(X_test)
26.
27.     # 计算预测的均方误差
28.     mse = mean_squared_error(y_test, y_pred)
29.
30.     # 预测未来7天的销售额
31.     next_7_days = np.array([max(category_data['day_num']) + i for i in range(1, 8)]).reshape(-1, 1)
32.     future_forecast = model.predict(next_7_days)
33.
34.     return future_forecast, mse
35.
36. # 预测每个类别的销售额
37. forecasts = {}
38. errors = {}
39. for category in daily_sales['分类名称'].unique():
40.     forecasts[category], errors[category] = forecast_sales(category)
41.
42. # 打印每个类别的预测销售额和均方误差
43. for category in daily_sales['分类名称'].unique():
44.     print(f"类别: {category}")
45.     print("预测销售额:", forecasts[category])
46.     print("均方误差:", errors[category])
47.     print()

```

问题二结果：

日期	花菜类日补货 总 (千克)	花菜类定价策 (%)	花叶类日补货 总 (千克)	花叶类定价策 (%)	辣椒类日补货 总 (千克)	辣椒类定价策 (%)
2023-7-1	53.354	50.06%	211.456	33.25%	75.144	75.14%
2023-7-2	20.122	25.25%	260.457	27.43%	108.957	108.95%
2023-7-3	41.145	50.15%	261.424	42.88%	109.551	109.55%
2023-7-4	39.265	87.36%	164.145	20.38%	43.092	43.09%
2023-7-5	50.924	41.12%	262.481	91.23%	110.027	110.02%
2023-7-6	31.45	39.27%	260.473	38.74%	108.927	108.92%
2023-7-7	29.175	52.17%	203.124	55.48%	70.087	70.08%

日期	茄类日补货总 (千克)	茄类定价策 (%)	食用菌日补货 总 (千克)	食用菌定价策 (%)	水生根茎日补 货总 (千克)	水生根茎类定 价策 (%)
2023-7-1	9.791	96.00%	65.201	33.06%	21.827	84.08%
2023-7-2	8.732	41.80%	66.912	100.03%	11.628	44.21%
2023-7-3	5.622	64.84%	31.853	107.01%	15.661	88.24%
2023-7-4	10.835	105.51%	49.309	86.00%	9.709	32.74%
2023-7-5	3.295	95.34%	46.763	92.50%	18.394	52.91%
2023-7-6	6.624	109.11%	58.331	91.30%	10.224	33.92%
2023-7-7	6.153	84.10%	61.965	62.41%	14.355	38.11%

附录 4

介绍：问题三代码

```
1. import pandas as pd
2. import matplotlib.pyplot as plt
3.
4. # 读取 Excel 文件
5. file_path = 'C:/Users/Desktop/茄类.xlsx'
6. df = pd.read_excel(file_path)
7.
8. # 提取销售单价和销量数据
9. 销售单价 = df['销售单价(元/千克)']
10. 销量 = df['销量(千克)']
11.
12. # 将纵坐标范围限制在 0 到 8 之间
13. plt.figure(figsize=(10, 6)) # 设置图表大小
14. plt.scatter(销售单价, 销量, marker='o', color='darkblue') # 绘制散点图, 颜色设置为深蓝色
15. plt.title('Relationship between Unit Price and Sales Quantity') # 设置图表标题
16. plt.xlabel('Unit Price (CNY/kg)') # 设置横坐标标签
17. plt.ylabel('Sales Quantity (kg)') # 设置纵坐标标签
18. plt.ylim(0, 8) # 设置纵坐标范围为 0 到 8
19. plt.grid(True) # 显示网格线
20. # 保存图表到指定目录
21. plt.savefig(r'C:/Users/Desktop/茄类的销量与销售单价的关系图.png')
22.
23. # 显示图表
24. plt.show()
```

销售和价格之间的关系：

```
1. import pandas as pd
2. import matplotlib.pyplot as plt
3.
4. # 读取 Excel 文件
5. file_path = 'C:/Users/Desktop/茄类.xlsx'
6. df = pd.read_excel(file_path)
7.
8. # 提取销售单价和销量数据
9. 销售单价 = df['销售单价(元/千克)']
10. 销量 = df['销量(千克)']
11.
12. # 将纵坐标范围限制在 0 到 8 之间
13. plt.figure(figsize=(10, 6)) # 设置图表大小
14. plt.scatter(销售单价, 销量, marker='o', color='darkblue') # 绘制散点图, 颜色设置为深蓝色
```

```

15. plt.title('Relationship between Unit Price and Sales Quantity') # 设置图表标题
16. plt.xlabel('Unit Price (CNY/kg)') # 设置横坐标标签
17. plt.ylabel('Sales Quantity (kg)') # 设置纵坐标标签
18. plt.ylim(0, 8) # 设置纵坐标范围为 0 到 8
19. plt.grid(True) # 显示网格线
20. # 保存图表到指定目录
21. plt.savefig(r'C:/Users/Desktop/茄类的销量与销售单价的关系图.png')
22.
23. # 显示图表
24. plt.show()

```

问题三结果：

单品编码	日补货总量 (KG)	定价策略 (%)	单品编码	日补货总量 (KG)	定价策略 (%)
102900005115625	3.172	111.79%	102900011035078	2.762	102.07%
102900005115779	10.443	134.23%	102900011035740	1	85.37%
102900005115878	9.014	142.23%	102900011021842	7	71.14%
102900005115861	18.215	94.02%	102900011023464	5.317	127.20%
102900005116257	2.299	46.21%	102900011030059	13	98.47%
102900005116530	17.72	47.46%	102900011030097	9	94.31%
102900005116714	18.037	60.12%	102900011030110	15	137.31%
102900005116899	24.97	128.36%	102900011030905	1	63.44%
102900005122654	2.129	59.75%	102900011031100	19	118.59%
102900011013274	2	125.27%	102900011031216	13	118.18%
102900011016701	26.919	58.49%	106949711300259	20	74.51%
102900011033944	3.151	138.72%	102900051010455	7.795	96.43%
102900011034231	13	70.94%	102900051000463	4.015	38.87%
102900011034330	10	53.00%	102900011008164	6.471	36.31%
102900011034439	5	59.37%	102900011030034	15	93.41%

附录 5

介绍：问题四代码

```
1. import pandas as pd
2. import seaborn as sns
3. import matplotlib.pyplot as plt
4.
5. # 文件路径列表
6. file_paths = [
7.     "C:/Users/Desktop/分析结果/花菜类.xlsx",
8.     "C:/Users/Desktop/分析结果/花叶类.xlsx",
9.     "C:/Users/Desktop/分析结果/辣椒类.xlsx",
10.    "C:/Users/Desktop/分析结果/茄类.xlsx",
11.    "C:/Users/Desktop/分析结果/食用菌.xlsx",
12.    "C:/Users/Desktop/分析结果/水生根茎类.xlsx"
13. ]
14.
15. # 创建一个空的 DataFrame 来存储数据
16. data = pd.DataFrame()
17.
18. # 从每个文件中读取数据并添加到 DataFrame
19. for file_path in file_paths:
20.     class_name = file_path.split('/')[-1].split('.')[0] # 获取类别名称
21.     df = pd.read_excel(file_path)
22.     data[class_name] = df['销售量'] # 假设销售量的列名为"销售量"
23.
24. # 计算各品类销售量之间的相关系数
25. corr_matrix = data.corr()
26.
27. # 清除 Matplotlib 缓存并重新加载配置
28. plt.rcParams.update(plt.rcParamsDefault)
29.
30. plt.rcParams['font.family'] = 'SimHei'
31.
32. # 创建分类名称的英文缩写字典
33. abbreviation_dict = {
34.     "花菜类": "CV",
35.     "花叶类": "FLV",
36.     "辣椒类": "PV",
37.     "茄类": "EV",
38.     "食用菌": "EF",
39.     "水生根茎类": "ARSV"
40. }
```

```
41.
42. # 替换标签文本
43. corr_matrix.rename(index=abbreviation_dict, columns=abbreviation_dict, inplace=True)
44.
45. # 自定义颜色映射, 包含一点点颜色变化
46. custom_cmap = sns.color_palette(['#F5F5F5', '#E8E8E8', '#D0D0D0', '#C2C2C2', '#B0B0B0', '#A0A0A0', '#909090'])
47.
48. # 可视化相关系数矩阵, 使用自定义颜色映射
49. sns.set(style='white')
50. fig, ax = plt.subplots(figsize=(10, 8))
51. sns.heatmap(corr_matrix, annot=True, cmap=custom_cmap, ax=ax)
52. ax.set_title('Correlation coefficient matrix of sales volume for various categories of vegetable')
53. plt.xlabel('Classification abbreviation')
54. plt.ylabel('Classification abbreviation')
55.
56. # 保存图表到指定路径
57. save_path = "C:/Users/Desktop/六大类蔬菜相关系数矩阵图.png"
58. plt.savefig(save_path)
59.
60. plt.show()
```

周铠然 · 仅供求职展示